

"Android" and the GPL license: Is the Linux- adaptation by Google still free software?

Leibniz Universitat Hannover

EULISP Program

AUTHOR: Luis Enríquez Alvarez

DATE: 12.01.2013



**“This work is licensed under the Creative Commons Attribution 3.0 Unported License. Follow the link below for terms and conditions:
<http://creativecommons.org/licenses/by/3.0/legalcode>”**

“INFORMATION SHALL BE FREE”

CONTENTS

1. INTRODUCTION	1
2. GNU, LINUX AND GPL LICENSE	
2.1. GNU and Linux are different.....	1
2.2. GNU General Public License and Copyleft	2
3. THE ANDROID OPERATING SYSTEM LICENSE PARADIGM	
3.1. Android and the preference of Apache License 2.0.	4
3.2. The Apache license 2.0 and GPL compatibility.	6
4. ANDROID IS LINUX BUT NOT GNU	
4.1. Technical Approach.....	7
4.2. Legal Approach.....	7
4.3. Philosophical Approach.....	7
5. ANDROID V. FREE SOFTWARE PRINCIPLES	
5.1. NO freedom to run the program for any purpose.....	8
5.2. Freedom to study how the program works and change it.....	8
5.3. Freedom to redistribute copies.....	9
5.4. Uncertain freedom to redistribute copies of modified versions.....	9
6. ANDROID BY GOOGLE AND OPEN SOURCE	
6.1. Free Software and Open Source Software philosophical difference..	11
6.2. Android by Google and Open Source Software principles.....	11
7. UNOFFICIAL VERSIONS OF ANDROID AND FREE SOFTWARE PROJECTS	
7.1. Unofficial versions (CyanogenMod).....	13
7.2. Free Software projects.....	14
8. CONCLUSIONS	14
9. BIBLIOGRAPHY	15
ABOUT THE AUTHOR	17

1. INTRODUCTION

In the last years Free Software has become a powerful alternative for users, enterprises and governments. The traditional copyrights theory was suddenly challenged by the development of the GPL license with the appearance of the Linux kernel¹, and the released of the GNU/Linux system. As a result, and with the gain of popularity of free software, today's world has become a very complex soup of free and proprietary licenses making very difficult to determine the legal nature of software, and new operating systems.

The Android operating system for since their first beta version in 2007, has gained a lot of popularity, and one of its strengths is their reputation as a *free operating system*, in contrast with Apple OS, or Windows. However, Android has been widely criticized by Free software activists and communities basing their arguments on the facts that android does not follow Free Software fundamental principles.

The purpose of this seminar is to examine the nature of the GPL license and if the Android operating system meets the requirements for being considered free software. What is free software? Is Linux just a kernel? Is GNU the same than Linux? Does Android follow these principles? Why did Android reject the use of GPL license? Is the GPL license strictly necessary for Free software? Is Android Free Software or just Open Source? Can unofficial versions of Android become Free Software? These and many other other questions will be confronted in the next pages.

2. GNU, LINUX AND GPL LICENSE

2.1. GNU and Linux are different. Let's start with the basics, GNU and Linux are related but not the same. GNU means *GNU not Unix*, and is a collection of software tools such as libraries, compilers, debuggers, that integrate an operating system. An Operating System has to be understood as the collection of tools that permits controlling the hardware components of a computer, and also is the essential layer in which software applications operate. The GNU operating system has been developed by the GNU project since 1984 and with no doubt, the most representative GNU developers is Richard Stallman. He founded the Free Software Foundation² in 1985.

It is relevant to consider that the GNU project started the development of a Kernel called *GNU Hurd*³ around those years. Considering that the kernel is the essential part of an operating system, in order to communicate data processing applications with hardware

1 Kernel is the bridge between applications and the actual data processing done at the hardware level. See, [http://en.wikipedia.org/wiki/kernel_\(computing\)](http://en.wikipedia.org/wiki/kernel_(computing)).

2 See, <http://fsf.org>.

3 See, <http://gnu.org/software/hurd/hurd.html>.

components, the Hurd kernel was a project in order to create a genuine and complete Free Software operative and kernel called GNU/Hurd.

Linux is a *Kernel* developed by Linus Torvalds in 1991 and appeared a lot earlier than the GNU/Hurd was expected to be released. It was a very good idea to combine efforts to put the Linux kernel into real work using the GNU operating system, so the result was *GNU/Linux*. The Linux kernel was dependent on the GNU system, and GNU system was dependent on the Linux Kernel, so the system was launched in 1992. However, Linus Torvalds motivation for the using of GNU system seems to be that he had not other choice, as he mentioned: “*Sadly, a kernel by itself gets you nowhere. To get a working system you need a shell, compilers, a library etc*”⁴.

So basically, Linux is a kernel, and GNU an Operating System. Computer science people know very well the difference, but after a while, users and then developers started calling the the GNU/Linux Operative System just Linux, forgetting about the GNU part. But in fact, most Linux distributions of today are based on the GNU system. For example, for Ubuntu Canonical's distribution, Ubuntu would be the operating system distributed by Canonical, and is totally based in GNU/Linux system, because it uses GNU's OS core components like the glibc⁵, and Linux kernel in order to operate. E.g. You can certify all this just by typing in the terminal the command *uname* with the *-o* option.

There are other distributions using the GNU operative system but with other kernel, like the *GNU/KFreeBSD*⁶, or other operative systems that run Linux kernel, and one of them seems to be the *Android OS*. Android OS does not use GNU operating system but uses the Linux kernel, so the result would be Android/Linux. The Android/Linux situation will be expanded in the next chapters.

2.2. GNU General Public License and Copyleft. The GPL License was created by Richard Stallman and it emerges as a result of all developments of the GNU project in the 80s. An important event that led Stallman to create the license was a conflict with *Unipress*⁷, enterprise which bought James Glossing rights about some C libraries for Emacs⁸. Stallman then created GNU Emacs by replacing those libraries with others, and decided to create a legal document to prevent free code from being proprietary⁹. A clear predecessor of the GPL is the Emacs General Public Licence in 1988.

In 1989 the GPL license came into existence, and GNU/Linux was released under GPLv2 license in 1992. The GPL license guaranties the freedom of users to copy, distribute, study, change and improve the software. For the Free Software Foundation “*free software is a matter of liberty and not price*”. In order to get these objectives, a distribution of the source code is fundamental.

4 See, http://en.wikipedia.org/wiki/GNU/Linux_naming_controversy.

5 GNU C Library. Is used as the C library in the GNU systems and most systems with the Linux kernel. Ref: <http://gnu.org/software/libc>.

6 See, <http://debian.org/ports/kfreebsd-gnu>.

7 Unipress emacs no longer exists. It was replaced by GNU Emacs. See, <http://emacswiki.org/emacs/jfm3>.

8 Powerful text editor developed by Richard Stallman. See, <http://gnu.org/software/emacs>.

9 See, http://free-soft.org/gpl_history.

The philosophy behind free software was based on the principle of copyleft¹⁰. Copyleft means that all products and applications that uses free software, should also be free software, therefore use the GPL license. The GPL v2 license establishes: “...*You may not copy, modify, sub-license, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sub-license or distribute the Program is void, and will automatically terminate your rights under this License*”¹¹. This prohibition makes clear that any work derived from GPL has to be GPL. Nevertheless, not all GNU licenses have this strong copyleft, and considering that most libraries may require interoperability with others with weaker copyleft, or no copyleft, the free software foundation created other license with weak copyleft, the *Lesser GPL*¹².

Free software and open source software are often referred as the same by most users, and from a legal point of view, they are. The *open source* emerged in 1998, and should be understood as a business term, focused in the development of code from a practical approach, and not *freedom* as an ethical approach. For Richard Stallman, the difference is philosophical¹³.

Now the first question arises, Is copyleft a fundamental condition for free software? Under the Free Software Foundation definition, four freedoms are essential for free software¹⁴:

- Freedom to run the program for any purpose.
- Freedom to redistribute copies.
- Freedom to modify the software and distribute modified versions with the source code.
- Freedom to study how the program works and change it. Access to the source code is needed.

Nevertheless, the Free Software Foundation(FSF) specifies that there are other Free software licenses and some of the them do not belong to the FSF. Some of these other Free software licenses may have a weak copyleft e.g. the Lesser GPL License(LGPL), the Mozilla Public License¹⁵(MPL), the Common Development and Distribution License(CDDL)¹⁶. However other licenses do not have copyleft e.g. the Apache Software License(ASL)¹⁷ or the BSD License¹⁸.

From this comparison, we can certainly establish that all GNU software is Free Software, but not all Free Software is GNU. From this assertion we can determine that

10 Copyleft is a general method for making a program(or other work) free, and requiring all modified and extended versions of the program to be free as well. <http://gnu.org/copyleft>.

11 Ref: GPL v2 license <http://gnu.org/licenses/gpl-2.0.html>.

12 See, <http://gnu.org/licenses/lgpl>.

13 See, <http://gnu.org/philosophy/free-software-for-freedom.html>.

14 See, <http://www.gnu.org/philosophy/free-sw.html>.

15 See, <http://mozilla.org/MPL>.

16 See, <http://opensource.org/licenses/CDDL-1.0>.

17 See, <http://apache.org/licenses/LICENSE-2.0.html>.

18 See, <http://opensource.org/licenses/BSD-2-Clause>.

any programmer can create free software, if the basic freedoms are granted. Nevertheless, the complexity of software systems in our days, makes very difficult to license a product by the GPL license because different licenses may apply over different components of the software like libraries, debuggers, plug-ins, compilers, kernels, amongst others. Under this perspective, developers need to understand in detail the restrictions of the GPL license, and the viral effect of copyleft. In the end, understanding the synchronicity of licenses is crucial to avoid confusion.

3. THE ANDROID OPERATING SYSTEM LICENSE PARADIGM

Android is a Linux-based Operating system developed by *Android Inc*¹⁹. Android Inc was supported and then bought by *Google Inc*. It was released in 2007 in cooperation with the *Open Handset Alliance*²⁰. Android has become a huge success, and its portability makes it suitable for smartphones, tables, net-books, and even smart TVs. The Android system is used today in more than 150 million devices. It is suitable to say, that despite the success of Linux distributions in Servers, and the relative success of Ubuntu and Fedora distributions in personal computers, the Android OS is the first Linux kernel distribution that reaches such amount of users. At the moment is the most popular OS for mobile devices, and it is expected that soon will be used in more devices than Windows OS. So the importance and Android OS in our days is crucial.

3.1. Android and the preference of Apache License 2.0. Android was released under the *Apache Software License 2.0*, a well known Free software license with no copyleft. Copyleft is an essential element of the GPL license, but not of all free software licenses. The ASL 2.0 is an example of a Free Software license with no copyleft.

The nature of ASL 2.0 is very simple to determine. Apache is an open source HTTP Web server developed and supported by the Apache Software Foundation. Considering that Apache is a platform for the development of Web Sites, it permits commercial use of applications under the terms of the Apache license. The copyleft element of GPL license would be an obstacle for the development of the website development model.

Now, why would Android OS prefers the ASL 2.0 over other Free Software licenses like the GPL, or even LGPL? The Android developer team justifies their choice with these arguments:

“Android is about freedom and choice. The purpose of Android is to promote openness in the mobile world, but we don't believe it's possible to predict or dictate all the uses to which people will want to put our software. So, while we encourage everyone to make devices that are open and modifiable we don't believe it is our place to force them to do so...”²¹

From a Free Software philosophical perspective, there is a very questionable argument in this explanation: *Is against freedom and choice forcing people to use free*

19 See, http://elinux.org/Android_History.

20 See, <http://openhandsalliance.com>.

21 Ref: <http://source.android.com/source/licenses.html>.

applications?, and if that is the case, *Where and by whom are they forced to do that?*

The arguments for not preferring the LGPL license are the following:

- Restricts OEM²² designs.
- Requires allowance of customer modification and reverse engineering for debugging those modifications.
- Large number of compliance problems for downstream device makers and application developers.

OEM designs is a powerful argument considering that mobile devices deal with different hardware manufacturers. The second argument can be considered a limitation of free software fundamental principles that is *modification* of the software. The third argument stands on the favor of developers.

Now, for most free software communities the reason why android adopted the ASL 2.0 is quite obvious, it allows the development of commercial applications. The LPGL would still allow the creation of commercial applications, but there is still a weak copyleft on it. In my own perspective, the main reason why Android rejected the use of the LGPL is to avoid the copyleft, as it is written at the end of the third argument: “...it is most prudent to simply not use LGPL libraries if we can avoid it”²³

Let's analyze the most important Apache License 2.0 prescriptions:

- Derivative Works. The Apache license 2.0 is very open about their derivative works approach: “...for the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link(or bind by name) to the interfaces of, the Work and Derivative Works thereof”²⁴. This prescription is very suitable for modern software in which plug-ins and derived works can be licensed under another license.

- No copyleft. “You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications or for any such Derivative Works, as a whole...”²⁵ This prescription allows sub-licensing, and the possibility of changing license terms.

However, we cannot blame Google for not preferring the GNU licenses. The philosophy of Google is just not compatible with the GNU philosophy. From a legal perspective it does not have to be compatible, because as we already established, the Android OS is not a derivative work of the Linux kernel, so it does not have to use the GPL license.

Now, let's start mention Linus Torvalds words: “...Android's current situation is no different than Red Hat or Suse situations ten years ago ... and that is the way development happens... Google has one hundred million devices running their code, they must be doing something right”²⁶. There is a powerful argument called development here, and it is obvious that every company develops in their own way. If

22 Original equipment manufacturer. http://en.wikipedia.org/wiki/Original_equipment_manufacturer.

23 Ref: <http://source.android.com/source/licenses.html>.

24 Transcribed from the Apache License 2.0, section(1).

25 Transcribed from the Apache License 2.0, section(4).

26 Linus Torvalds, from Linux Conference 2012 in Barcelona. See, <http://linux.com/news/software/applications>.

we take for example Red Hat Linux created ten years ago the model of *Enterprise Linux*, which consists in offering support to enterprises and charge a small fee for the services. The result was that Red Hat distributions have become very respected in the market because of their support to the customer, but still free software.

Despite that Linux Torvalds has disappointed many Free Software activists, because of his attitude towards Google, he defends the freedom of Google about the development and evolution of Android. If it is Free Software, just Open Source, or maybe becomes Proprietary Software in the future will be in the hands of Google Inc.

3.2. The Apache license 2.0 and GPL compatibility. As we have seen, we cannot blame Google for not preferring the GNU licenses. The philosophy of Google is not compatible with the GNU philosophy. But from a legal perspective, Does it have to be compatible?

As we have seen, despite the use of the Apache license for Android OS, the Android OS viewed as a whole also uses GPL license. The Linux kernel is licensed under GPL v2 license, so Android distributions are using GPL v2 license. If we consider that GPL includes a strong copyleft we might wonder why Android OS was free to choose another license than GPL. From a general perspective, the GPLv2 of Linux kernel is a license with strong copyleft, and therefore it would not be compatible with the ASL.

However, this is not always the case. The answer is that the applications don't have to follow the licensing of the kernel *if they are not linked to the kernel*. This is a gray area to understand and there are arguments in favor and against. In one hand, GPL defenders argument that the Android OS should be licensed under GPLv2 because it runs with the Linux kernel. In the other hand Android defenders argument that the kernel interacts with the Android OS just by exchanging normal system calls. There are two keys to decipher this controversy:

(1) Is Android OS a derivative work of Linux kernel?. The answer is NO. A kernel just interacts with the operating system by providing a hardware abstraction and exchanging system calls with the purpose of providing necessary services for running applications²⁷.

For example, The *file system* is an illusion created by the kernel, and its purpose is organizing our data at the logical level of the Operative System, so when we store a file in the folder *Desktop*, there is a Desktop at the logical level, but not at the physical level. At the physical level this file would be stored in sectors that can be mapped in different sizes like 512mb, and the data would be registered following different allocation algorithms.

From this perspective, the operating system and the kernel are different. They are dependent on each other, but they are not a derivative work of each other. The ultimate proof is to consider that the GNU operating system tools appeared before the appearance of the Linux kernel in 1991. So How could the GNU operating system be a derivative work of the Linux kernel?

²⁷ See, <http://tuxradar.com/content/how-linux-kernel-works> . It includes a well described methodology to understand the system calls of the Linux kernel.

(2) Linus Torvalds permission. Linus Torvalds is the copyright holder of the Linux kernel, and he allows anyone to use his kernel under simple conditions:

*“NOTE! This copyright does **not** cover user programs that use kernel services by normal system calls – this is merely considered normal use of the kernel, and does **not** fall under the heading of derived works. Also note that the GPL below is copyrighted by the Free Software Foundation, but the instance of code that it refers to (the Linux kernel) is copyrighted by me and others who actually wrote it²⁸”.*

Linus Torvalds distributes this note within the GPL v2 license, for the distribution of Linux kernels. He makes it clear that in the end he is the copyright holder of the Linux kernel, and not the Free Software Foundation. If the copyright holder authorizes the use of his product, there is not copyright infringement.

So in the end, the GPLv2 license will apply only to the Linux kernel and patches. Thus, the GPL v2 of the Linux Kernel, and The ASL 2.0 of Android are valid together, even if the licenses are not compatible.

4. ANDROID IS LINUX BUT NOT GNU

Now it is time to confront all the arguments presented in the previous lines. It is clear that *Linux* is a kernel, and *GNU* the system. It is also clear that not all free software licenses follow the FSF²⁹ principles. So, what kind of Frankenstein is Android?, let's confront a technical, a legal and a philosophical approach:

4.1. Technical Approach. Android is an operating system by itself. Android OS does not use the famous *glibc* library of the GNU system. The *glibc* in GNU/Linux is licensed under the LGPL license and defines the kernel system call interfaces, and therefore is essential to run the system. The Android OS uses their own creation called the *Bionic libc*³⁰ library, and released it under the BSD license. This library fulfills the purposes of the GNU's *glibc*. Thus, if there is not GNU system in it, there is not an infringement of the GPL, but of course to confirm this affirmation, a complete examination of all files must be required.

Assuming that there is not such GPL infringement, even if the code is proprietary software, and has been openly distributed, libraries are still proprietary. Therefore, the only part of Linux in Android is Linux itself, the kernel.

So basically, from a technical approach Android is Linux, but it is not GNU.

4.2. Legal Approach. The Android operating system is released under the Apache Software License 2.0, a Free Software license with no copyleft. However, the Linux kernel is licensed under GPL v2. As we have seen, there would not be a conflict between these licenses because Google's code is not a derivative work of the kernel.

28 Transcribed literally from <http://kernel.org/pub/linux/kernel/COPYING>.

29 Free Software Foundation.

30 The Bionic libc is a derivation of the BSD standard C library code that was originally developed by Google for the Android operating system. [http://wikipedia.org/wiki/Bionic_\(software\)](http://wikipedia.org/wiki/Bionic_(software)).

Then we can conclude that from a legal perspective android is not Linux, and is not GNU. Android is Google using ASL 2.0.

4.3. Philosophical Approach. Google philosophy is by far different than the GNU philosophy. Linux is a kernel, but it implicitly shares the GNU philosophy by being released under GPL license. As we have seen, this is not totally true because Linus Torvalds opinions are different than GNU perspectives. So from a philosophical perspective Android is not Linux, and is not GNU, but perhaps they are compatible with Linus Torvalds development perspective. In the end, Android follows Google's philosophy.

5. ANDROID V. FREE SOFTWARE PRINCIPLES

At this point it should be clear that Android OS is not GNU, Android OS is not GPL, it uses Linux kernel but is not forced to use GPL because it is not a derivative work of the Linux kernel. So now let's move on, and analyze if Android could still be considered Free Software, and in order to do that, I will compare the Android OS with the main four free software principles:

5.1. No freedom to run the program for any purpose. From a technical perspective, Android devices had came with several limitations by default.

The most controversial limitations are:

(1) Boot-loader locked. Most Android devices come with the boot-loader locked by the manufacturer, so it is not possible to run it for any purpose. Indeed, for any purpose that is not determined by the manufacturer. Basic functions like enhancing the functionality of the device, necessary recovery procedures, or the possibility of install a Modified version, are restricted by this measure. However, some companies like Asus, or Samsung have created an unlocked software, but ironically, the use of their own unlocking software, will finish the warranty of the device³¹.

(2) No root privileges. The restriction of root access is a clear violation of free software principles by Android OS, because without root privileges, it is not possible to install and run many applications. It is also not possible to do modifications of the system. It is also not possible to install modified versions, or other operative systems! I am sure that all Free Software users have felt disappointment in their first Android OS contact, because of these restrictions.

The process of rooting the device is not straightforward, as it is required first to unlock the device, then install a recovery ROM environment called the ClockworkMod³² recovery, and once this is done, finally root the device by installing a modified firmware³³.

As we have seen, Android OS does not fulfill the freedom of running the software for

31 The unlocked software depends on the device. Here is a guide to unlock a Verizon Samsung Galaxy S III smart phone: <http://xda-developers.com/android/verizon-samsung-galaxy-s-iii-unlocked>

32 See, <http://clockworkmod.com>

33 The process of Rooting Android devices depends on the device. Here is a guide to root a Samsung Transformer tablet. See, <http://androidegis.com/how-to/learn-how-to-root-the-asus-transformer-tf300t/>

any purpose.

5.2. Freedom to study how the program works and change it. The most popular firmwares³⁴ of Android until now have been: Gingerbread³⁵(2.3), HoneyComb³⁶(3.1-3.2), Icecream Sandwich³⁷(4.03-4.04) and Jelly Bean³⁸(4.1). The source code of android has been provided in most of its distributions, sometimes with considerable delays, and in the case of HoneyComb, it was not complete. But the fact is that it has been provided, and that is the reason why new firmwares and applications have been developed. A very popular alternative firmware for android devices is *CyanogenMod*³⁹ firmware, because it replaces the proprietary firmwares from manufacturers, and allows rooting the device. There are different versions of CyanogenMod, based in different android firmwares. In any case, Android OS fulfills this freedom, because it has mostly distributed the source code, with some delay exceptions.

5.3. Freedom to redistribute copies. The fact that many proprietary manufacturers like Samsung or Asus are distributing their own android copies, shows that this freedom exists. Also, there is not limitation for distributing Android OS in community projects like the *Rowboat project*⁴⁰. Or the *XDA developers community*⁴¹. So far, Android by Google fulfills this freedom.

5.4. Uncertain freedom to redistribute copies of modified versions. There is not legal infringement if modified versions are distributed. However, Google seems to dislike them. That is the case of the CyanogenMod⁴² distribution. But first, in order to understand freedom of modified distributions, we should first understand how the mess of distributions works. All systems that use the Linux kernel are known as Linux distributions, but Android is an operative system, not a distribution. Could it be considered as a Linux Distribution? Let's compare with other Linux distributions:

To clarify this, let's assume that there are different distributions. E.g. Ubuntu⁴³ is a trademark of Canonical, was based on Debian⁴⁴, which distributes the GNU operative system. The same with Fedora⁴⁵ it belongs to the Fedora's project, was based on Red Hat⁴⁶, and it also distributes GNU/Linux, but with some differences. So from this perspective Android is a trademark of Google, it is an operative system, and distributes

34 Firmware is the combination of persistent memory and program code and data stored in it.

<http://wikipedia.org/wiki/Firmware>.

35 See, <http://developer.android.com/about/versions/android-2.3-highlights.html>.

36 See, <http://developer.android.com/about/versions/android-3.0-highlights.html>.

37 See, <http://developer.android.com/about/versions/android-4.0-highlights.html>.

38 See, <http://developer.android.com/about/versions/jelly-bean.html>.

39 CyanogenMod Is an after market firmware for a number cell phones based on the open-source Android operative system. <http://cyanogenmod.org>.

40 See, <http://code.google.com/p/rowboat>.

41 See, <http://xda-developers.com>.

42 CyanogenMod is an after market firmware distribution of Android OS. See, <http://cyanogenmod.org>.

43 See, <http://ubuntu.com>.

44 See, <http://debian.org>.

45 See, <http://fedoraproject.org>.

46 See, <http://redhat.com>.

the Android OS system.

So in this area, how many *Androids* can we get other than the developer of android? Could *CyanogenMod* be considered an alternative distribution of Android? Let's make a comparison:

From a proprietary software perspective, there is usually just one official distribution, and that is it. For example, OSX is an operative system that belongs and is distributed by *Apple Macintosh*. Non official distributions can exist, but they would be considered illegal because they are breaking proprietary copyrights. The *Hackintosh* distribution would be illegal and its products are illegal considered modified works of the OSX system.

From a Free software perspective there are two scenarios:

- Many distributions can co-exist when their license contains strong copyleft, like Ubuntu from Debian, or Linux Mint⁴⁷ from Ubuntu.
- There is only an official distribution, but other distributions are allowed if they meet certain criteria. Such as OpenBSD⁴⁸ or FreeBSD⁴⁹ in the BSD operative system.

So in the case of Android, there is not legal infringement against Google, but CyanogenMod is not considered official! very interesting. The last controversial point is that Google does not permit CyanogenMod to use proprietary Google applications in their distributions. We will continue this confrontation in the next chapter.

After these principles confrontation we can establish that Android by Google doesn't fulfill two of the four essential free software principles. So from a radical Free Software Foundation perspective, *Android by Google is not Free Software*. In one hand Richard Stallman has mentioned openly that Android is less evil than Apple Macintosh and Windows, but doesn't follow free software philosophy⁵⁰. On the other hand, Linus Torvalds has mentioned that he doesn't see the difference between Android and Red Hat Linux or Open Suse⁵¹ ten years ago⁵². Does this mean that is not free software, but it can become free software after a while? Can some distributions of Android be Free Software, and others not?

6. ANDROID BY GOOGLE AND OPEN SOURCE

As we have mentioned earlier, the difference between *free software* and *open source software* is very narrow, and come from a philosophical perspective. Free software is focused on fundamental rights, and Open Source is more about functionality and development. For Bruce Perens, "*the main purpose of Open Source is marketing the Free Software philosophy to business people who are more concerned with profit than freedom*"⁵³. Thus, Open Source is a functional category, and doesn't have to fulfill

47 See, <http://linuxmint.com>.

48 See, <http://openbsd.org>.

49 See, <http://freebsd.org>.

50 Richard Stallman to The Guardian UK. See, <http://guardian.co.uk/technology/2011/sep/19/android-free-software-stallman>.

51 See, <http://opensuse.org>.

52 Linus Torvalds, from Linux Conference November 2012 in Barcelona. See, <http://linux.com/news/software/applications>.

53 Bruce Perens is the leader of the Open Source movement. See, <http://perens.com>.

the four principles of Free Software.

Under this affirmation we can clearly establish two facts:

- Free Software has to be *open source*, but *open source* doesn't have to be *free software*. This principle applies to licenses.

- Open source software can be used for the development of commercial purposes.

The Open Source appearance in 1998 contributed to the appearance of new terms like *Free open Source Software (FOSS)*, and *Free libre and Open Source Software(FLOSS)*. These new terms have been widely accepted, but also, they bring more confusion to an already very narrow distinction.

6.1. Free Software and Open Source Software philosophical difference. At this point the philosophical distinction between Free Software and Open Source makes sense. The well known Richard Stallman phrase “*free as freedom and not as free*⁵⁴”, can actually be crucial. Many believe that the copyleft in the end creates a barrier between Free Software and Open Source Software, because it forces to derivative works to use GPL compatible licenses following the Free Software principles, with the purpose of defending the freedom of users.

So if we follow the comparison *all Free Software is Open Source, but not Open Source is Free Software*, then we must consider why there are Free Software licenses other than GPL that don't have copyleft. For E.g. BSD is a kernel, is an Operating system family, and is free software, because it follows the Free Software principles. But The BSD license doesn't have copyleft, so derivative works can have other license terms.

So in the end, the copyleft protects the developing of free software, and the future freedom of users, but it is not a condition for free software. Now, GNU and BSD software are *free software*, and therefore they are *open source software*, as well.

In the other hand, if open source software does not have to follow the free software principles, Why it can be licensed under a free software license? This is the case of Android OS. As we have seen it doesn't follow the free software principles, and in the end, it does not have to. But then, Why does it uses a free software license like the ASL 2.0?, and the BSD license for libraries?.

The answer is that free software licenses are also open source software licenses, and if they are compatible, there is not restriction. The philosophical difference has no legal effect, but makes a difference because the principles of open source software are philosophically different than the principles of free software.

6.2. Android by Google and Open Source Software principles. With all these in mind, we could think that if Android by Google doesn't meet the criteria for Free Software, it could be considered at least open source. So let's analyze if Android meets open source principles:

- **Free redistribution.** Android OS is free redistributed in terms of free of

54 See <http://gnu.org/philosophy/free-sw.html>.

royalties.

- **Source Code.** Source code is available, at least the major part.
- **Derived Works.** Apache license allows derived works and modifications.
- **Integrity of the Author's Source code.** Under this principle, the license must permit distribution of software build from modified source code but it may require different name and version number than the original software. Apache license meets this criteria.
- **No Discrimination against persons and groups.** There is not discrimination, at least in theory.
- **No Discrimination against fields of endeavor.** The same as above.
- **Distribution of license.** Distribution of Android by Google is granted.
- **License Must not be specific to a product.** Apache license is used by other products.
- **License must not contaminate other software.** It does not.
- **Example licenses.** Finally, the definition mentions as open source software licenses the GPL, and the BSD license. The Apache Software License is 2.0 is indeed an open source software license, too.

It is important here to analyze the open source licenses in the light of the facts prescribed at the beginning of this chapter. Free Software licenses are also Open Source licenses. But free software licenses have different categories according to the strong, weak, or no copyleft.

This is why in my opinion, *Android by Google* fulfills the criteria of Open Source software. The ASL 2.0 is a free software and open source license with no copyleft, and Android by Google is Open Source Software with no copyleft in it.

In addition, Google does not refer to Android as Free Software in their official platforms, they refer to Android as an *open source project*. The confusion has been brought by free software communities discussing about the fact that it is based in the Linux Kernel, and therefore it should be free software under the GPL v2 license. But as we have seen, Android is not GPL, does not have to be GPL, and from a philosophical perspective, it is not free software. Android by Google follows open source principles, therefore is open source software.

7. UNOFFICIAL VERSIONS OF ANDROID AND FREE SOFTWARE PROJECTS

ASL 2.0 grants the possibility of creating and distributing derivative works. The section 2 of ASL 2.0 establishes:

“Subject to the terms and conditions of this License, each Contributor hereby grants to you a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form”.

Legally, any unofficial version of Android OS is possible. The only special requirement for distribution of unofficial versions is established in section 4(b) and 4(c) of the ASL 2.0:

“You must cause any modified files to carry prominent notices stating that You changed the files... You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works”.

Technological developments are very fast, so modified versions of Android OS appeared by the year of 2010, and it was called CyanogenMod. Under a legal perspective CyanogenMod is not “unofficial”, because legally Google allows their distribution. But let's keep calling it unofficial just for the purpose of differentiation. The fact is that with the appearance of unofficial distributions of versions of the Android OS, or simply, a truly Free Software competitor, Google has a bigger challenge to confront these days. Let's the two possible scenarios:

7.1. Unofficial versions (CyanogenMod). CyanogenMod appeared in 2010 and is maintained by the CyanogenMod project. There are different stable and nightly firmwares that replace all Android's official firmwares. The last version is CyanogenMod 10.1, and is a successful replacement of the official Jelly Bean 4.1 firmware. CyanogenMod is an Android adaptation of the GNU system. Also, the GNU project recommends the use of CyanogenMod in Android devices, because it does respect the Free Software principles, so for the GNU and Richard Stallman, CyanogenMod is Free Software.

In the other hand, Google has shown that they do not like unofficial distributions of Android like CyanogenMod, by the fact of not authorizing the distribution of most applications within it. But these applications can still be downloaded after the installation of CyanogenMod.

As an experienced GNU/Linux and Android/Linux user, my opinion is that the conditions for the installation of CyanogenMod are not so different from the process of installing any GNU/Linux or BSD distributions in a Windows machine. Warranty finishes when you unlock the device and root the Android system, so in the end, you are by your own and at your own risk.

In addition, something very interesting is that CyanogenMod is licensed under GPL. In my personal opinion the strength of GPL is their philosophical approach about defending fundamental rights of users. So many Android users that share GNU philosophy are switching Android by Google, for CyanogenMod, or even installing other GNU based distributions like Ubuntu(with some limitations). But of course, they are minority.

The applications scheme is something different than the OS itself. Many application

markets have emerged other than the official *play store*, like the *F-droid*⁵⁵. So even if Google doesn't want to provide applications for unofficial developers, there are many other places to get the same or similar applications.

It is important to remark that even if Android by Google does not respect all Free Software freedoms, or that they are alternatives like CyanogenMod, in practice the Android Open Source project has a very strong spirit of development and sharing. There are many independent developers working and experimenting for the Android OS, like the XDA developers project. The main focus of this community is development, so it is not relevant in the end if Android is not GPL. Also we should consider that a very good motivation for the developing of new applications is the possibility of having a market impact, and for many developers, the viral effect of copyleft can be seen as an obstacle. So again, philosophy could make a difference between Free Software communities, and Open Source Software communities.

7.2. Free Software projects. There are several movements today that have the purpose of make Android free Software, or develop a truly Free Software OS for mobiles. A campaign of great relevance is the *Free your android* campaign⁵⁶. This campaign is an example of purely free software communities fighting for the principles of Free Software. This campaign has been focused in two arguments: open hardware, and defense of privacy.

- Open Hardware. Proprietary manufacturers are using Android, but still using their proprietary drivers and firmwares, because obviously they are not interested in giving users total freedom to operate their devices, due to commercial reasons. So they promote the *GTA04*⁵⁷ motherboard for running GNU/Linux in phones, but unfortunately there are still problems of proprietary drivers of components.

- Google and privacy. Free Software activists consider that Google doesn't respect the fundamental rights of users, specially their Right to privacy. This affirmation is easily verifiable by checking the *history's feature* of Google search engine which records all users on line activities⁵⁸, or the lack of respect for privacy of some applications like Google earth, or Google maps. This affirmation gets more relevance by considering that for many users Google is an informant of Governments and Corporations. But just by itself, *Google and privacy* would be a huge and interesting topic for another investigation.

8. CONCLUSIONS

- **GNU and Linux are not the same.** GNU is an Operative system and Linux is a kernel. Linux was released under the GPL v2 license, a license with strong copyleft.
- **GPL is not the only Free Software license.** There other Free Software licenses with weak, or not copyleft. Android uses the Apache license 2.0, a Free Software license

55 See, <http://f-droid.org>.

56 More information at: <http://fsfe.org/campaigns/android/android.html>

57 See, <http://wiki.openmoko.org/wiki/GTA04>

58 Anyone can use this feature after signing into a Google account. See, <https://history.google.com>

with no copyleft.

- **Android is Linux but not GNU.** Google Inc replaced the famous *glibc* of GNU for his own *Bionic libc*. The only Linux in Android is the kernel itself.
- **Android does not need to use GPL license.** The Operative System is not a Derivative Work of the kernel. Therefore, the fact that Linux kernel is released under a non compatible license is irrelevant.
- **Android is not Free Software.** There is not *Freedom to run for any purpose* in Android devices. They come with the boot-loader locked, and without root privileges. Unlocking the loader and rooting the devices breaks the warranty agreement.
- **Android by Google is closer to Open Source than Free Software.** Android principles and philosophy does not fulfill the fundamental Free Software freedoms, however it meets the Open Source principles.
- **Unofficial Distributions of Android could become Free Software.** The development of alternative modifications of Android is a fact, and some of these new Android distributions are meant to be really free, in concordance to the Free Software principles. CyanogenMod is an example of a modified version of Android.

9. BIBLIOGRAPHY

BOOKS:

- Williams Sam, *Free as in Freedom: Richard Stallman's Crusade for Free Software*, Beijing, O'Reilly, 2002.
- Phillips Edmond and Campbell Doris, *The software license unveiled*, New York, Orford University Press, 2009.
- Van den Brande, Coughlan, Jaeger, *The International Free and Open Source Software law book*, Open Source press, 2011.
- Smith Roderick, *Linux Professional Institute Certification study guide*, Canada, Wiley publishing, 2011.
- Bovet Daniel and Cesati Marco, *Understanding the Linux Kernel 3rd Edition*, United States, O'Reilly, 2006.

WEBSITES AND FORUMS:

- <http://source.android.com/about/philosophy.html>
- <http://www.kernel.org/pub/linux/kernel/COPYING>
- <http://www.tuxradar.com/content/how-linux-kernel-works>
- <http://www.tuxradar.com/content/how-compile-linux-kernel>
- <http://stackoverflow.com/questions/2773437/glibc-philosophical-question>
- <http://www.groklaw.net/article.php?story=20110915194531435>
- <http://www.serverwatch.com/server-trends/red-hat-no-regrets-about-moving-to-the-enteprise-model.html>
- <http://www.lockergnome.com/osx/2012/02/24/are-hackintosh-computers-legal/>
- <http://www.dailytech.com/Android+OS+in+the+Hot+Seat+as+Oracle+v+Google+Legal+Battle+Kicks+Off+Today/article24468.htm>
- <http://www.gnu.org/licenses/gpl-2.0.html>
- <http://www.gnu.org/copyleft/>

- <http://www.linux.org/article/view/what-is-linux>
- <http://www.cyanogenmod.org/>
- <http://www.threadwatch.org/node/5588>
- <http://www.xda-developers.com/>
- <http://www.apache.org/foundation/>

ABOUT THE AUTHOR



Luis Enríquez. Copyright, Copyleft and Media lawyer. I did my formal education in Information Technology and Intellectual property Law(LLM - Leibniz Universität Hannover – Germany), International Economic Law(LLM - Univesidad Andina Simón Bolívar – Ecuador), Digital Forensics and Ethical Hacking(CHFI && CEH - ECCouncil), Algorithm Composition and Sonology(Royal Conservatoire – The Netherlands)... And my informal education in file sharing sites and Internet Forums for the last 15 years.

As a FOSS lawyer and a computer security researcher, my mission is to work directly with software developers and digital artists, talking in their 'hi-tech' language and solving their legal problems. In the other hand, I also work with lawyers, talking in their 'legal' language, and solving their technical problems. Don't hesitate to contact me:

email: luisenriquez@geek4nongeeks.com luisenriquez@fossilawyers.org